Botnet Detection by Monitoring Group Activities in DNS Traffic

Hyunsang Choi, Hanwoo Lee, Heejo Lee, Hyogon Kim
Korea University
{realchs, hanwoo, heejo, hyogon}@korea.ac.kr

Abstract— **Recent malicious attempts are intended to get financial benefits through a large pool of compromised hosts, which are called software robots or simply "bots." A group of bots, referred to as a botnet, is remotely controllable by a server and can be used for sending spam mails, stealing personal information, and launching DDoS attacks. Growing popularity of botnets compels to find proper countermeasures but existing defense mechanisms hardly catch up with the speed of botnet technologies. In this paper, we propose a botnet detection mechanism by monitoring DNS traffic to detect botnets, which form a group activity in DNS queries simultaneously sent by distributed bots. A few works have been proposed based on particular DNS information generated by a botnet, but they are easily evaded by changing bot programs. Our anomaly-based botnet detection mechanism is more robust than the previous approaches so that the variants of bots can be detectable by looking at their group activities in DNS traffic. From the experiments on a campus network, it is shown that the proposed mechanism can detect botnets effectively while bots are connecting to their server or migrating to another server.**

I. Introduction

Explosive growth of the Internet provides much improved accessibility to huge amount of valuable data. However, numerous vulnerabilities are exposed and the number of incidents is increasing over time. Especially, recent malicious attempts are different from old-fashioned threats, intended to get financial benefits through a large pool of compromised hosts. This horrifying new type of threats that endanger millions of people and network infrastructure around the world. For example, they steals personal information which can lead to significant financial losses and simultaneously, used for delivering spam mails, and launching DDoS (Distributed Denial of Service) attacks.

A large pool of compromised hosts, called bots, communicate with a bot controller to coordinate the network of bots. Such a network is commonly referred to as a botnet. An attacker, called a botmaster, controls a botnet to perform various malicious activities. Recent attacks show that their intentions are to gain financial benefits from the attacks.

Most bots can perform a hybrid of previous threats engaged with a communication system. They can propagate like Internet worms, hide themselves from detection systems, and launch DDoS attack like DDoS attack toolkits. These crossbreed techniques make the botnet intelligent and hard to be handled through a security mechanism. One prominent characteristic of botnets is the use of command and control (C&C) channels. The main purpose of the channels is to deliver the commands of a botmaster. And today's botnets use the Internet Relay Chat (IRC) protocol [1], which is mainly designed for group communication in discussion forum called channels. But the channels are now used for the communication of a botnet among distributed bots and their controller.

Defending against botnets is a pressing problem that is still not well comprehended, though botnets first appeared several years ago. Former defense mechanisms focused on a particular symptom of bots or a signature of bot programs. Even though the studies were meaningful to develop better defense mechanisms, their approaches have intrinsic limits such as the ineffectiveness for detecting unknown bot programs which are a slight modification of an existing bot program or newly generated bot programs. Recent studies such as [2] on botnet measurements and their detection also have the same weakness for the variants of bot programs.

The main contribution of this study is the development of an anomaly-based botnet detection mechanism by monitoring group activities in DNS traffic. Botmaster constructs and manages his botnet in several steps and bots rally to (C&C) server at an early stage. Most of bots use DNS in rallying process and the DNS traffic have unique features which we define as group activity. The DNS traffic also appeared in other stages therefore, by using the group activity property of botnet DNS traffic, we can detect botnet. There are a few study which use DNS to detect the botnet and some of them used DNS redirection to monitor botnets. However, they are easily evaded when a botmaster knows them. Nonetheless, our approach does not need any DNS redirection and communication with any component of botnet.

We have developed the botnet detection mechanism with the following four steps. First, we found several features of botnet DNS traffic that is distinguishable from legitimate DNS traffic. Second, we defined the key feature of DNS traffic called group activity. Third, we developed an algorithm that differentiate botnet DNS query by using group activity feature. Last, we analyzed the algorithm to prove feasibility of our mechanism. The mechanism are an anomaly-based detection mechanism, so that we can detect botnet regardless of the type of bot and botnet. The mechanism uses the information of IP headers and that enables to detect botnet, even though

This work was supported in part by the ITRC program of the Korea Ministry of Information, and Basic Research Program of the Korea Science & Engineering Foundation. This work was partially supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract(UD060048AD).

they uses SSH(Secure Shell) or any other channel encryption methods. Moreover, mechanism can detect botnet irrespective of protocol which they use. We also developed a mechanism that enable to detect C&C server migration. Botnet frequently change its C&C server by migrating to candidate C&C server. Our algorithm can find the botnet even though bots are migrating to other candidate C&C server.

Section 2 shows the related works of botnets. Section 3 describes main features of botnets, including the unique pattern of botnet DNS traffic, rallying problem and migration of botnet. Then, we will introduce a botnet detection mechanism in Section 4 and evaluate the feasibility and effectiveness of the mechanism in Section 5.

II. Related Work

The existence of botnets was recognized several years ago, but the studies for defending botnets are still in an early stage. Some security companies and institutions have analyzed the botnet traffic, the method of propagation and furthermore proposed the botnet detection and response mechanisms. However, their defense mechanisms are focused on the symptoms of abnormal network traffic and bot binary detections by matching with the signatures of known bot codes. Even though these are useful for many cases, they have inevitable limitations such that they are unable to detect new or modified bots.

There have been a few researches on the methodological analysis about the bot and botnet such as their behaviors, statistics, and traffic measurements. Jones [3] provided botnet background and recommendations so that network and system security administrators can recognize and defend against botnet activity. Cooke *et al.* [4] outlined the origins and structure of bots and botnets, data from the operator community and study the effectiveness of detecting botnets by directly monitoring IRC communication or other command and control activity and show a more comprehensive approach is required. Barford *et al.* presented a perspective based on an in-depth analysis of bot software source code and reveals the complexity of botnet software, discusses implications for defense strategies based on the analysis [5]. Rajab *et al.* [2] constructed a multifaceted infrastructure to capture and concurrently track multiple botnets in the wild, and achieved a comprehensive analysis of measurements reflecting several important structural and behavioral aspects of botnets. They studied the botnet behavior, botnet prevalence on the Internet, and modeling the botnet life cycle.

Recently, a few attempts have been made to cope with botnet problems and most of them have come to focus on detection of botnet. Bots are sending DNS queries in order to access the C&C channel server. If we could know the name of domain name of C&C channel server then we can blacklisting the domain name for sinkhole techniques to capture the botnet traffic and measure the botnet. Dagon *et al.* [6] identified key metrics for measuring the utility of a botnet, and describe various topological structures botnet may use to coordinate attacks. And using the performance metrics, they consider the ability of different response techniques to degrade or disrupt botnets. Their study used DNS redirection to monitor botnets. However our approach does not need any DNS redirection and communication with any component of botnet. Dagon also present botnet Detection and response approach [7] with analyzing peculiarity of botnet rallying DNS traffic (particularly, measuring canonical DNS request rate and DNS density comparison). However the detection technique could easily be evaded when botmasters know the mechanism and poisoned by using faked DNS queries. Kristoff [8] also suggested a similar approach, but the mechanism has the same weakness.

Binkley [9] proposed an anomaly-based algorithm for detecting IRC-based botnet meshes. The algorithm combines an IRC mesh detection component with a TCP scan detection heuristic called the TCP work weight. They can detect IRC channel with high work weight host but some of them could not be a member of botnet (false positive), additional analysis for many borderline cases as they mentioned in the paper. Ramachandran [10] developed techniques and heuristics for detecting DNSBL reconnaissance activity, whereby botmasters perform lookups against the DNSBL to determine whether their spamming bots have been blacklisted. This approach of botnet detection is derived from novel idea that detect DNSBL reconnaissance activity of botmaster but also have false positives and some defects that is referred in their paper.

Botnets are constructed and managed in several stages such as bot infection, C&C server rallying, and other types of malicious activities. Defense against botnet attacks seems to be a very complicated task. Only a few of works have been done in this area, but we need further improvements for the purpose of practical use. Moreover, previous works are difficult to be used for finding all types of botnet because the botnet have complex behavior patterns.

III. Botnet

A. Growth of Botnet

A botnet is a large pool of compromised hosts that are controlled by a botmaster. Recent botnets use the Internet Relay Chat (IRC) server as their C&C server for controlling the botnet. Botmaster can disperse commands to his botnet by the use of the IRC C&C channel. It was shown that most botnets use the IRC for C&C process [11], however the traffic among bots, the C&C sever and the botmaster can be considered as legitimate traffic because it is hard to distinguish from normal traffic.

The size and prevalence of the botnet reported as many as 172,000 new bots recruited every day according to CipherTrust [12], which means about 5 million new bots are appeared every month. Symantec [13] recently reported that the number of bots observed in a day is 30,000 on average. The total number of bot infected systems has been measured to be between 800,000 to 900,000. A single botnet comprised of more than 140,000 hosts was found in the wild and botnet driven attacks have been responsible for single DDoS attacks of more than 10Gbps capacity [14].

B. Rally Problem and IRC Server

Since vulnerable hosts are infected through self-propagating worms, email messages, messengers and other random spreading processes, the key problem of a botmaster is how to rally the infected hosts. Botmaster want their botnets to be invisible and portable and therefore, they uses DNS for rallying. It is possible to use other method for rallying the bots, however most of them cannot provide both mobility and invisibility at the same time. For example, if bot binary has the IP address of C&C server as hard coded string, then the C&C server can be perilous to reverse engineering. Even though the IP address of C&C server is obfuscated to prevent trivial reverse engineering analysis, the hard coded IP address is unchangeable, so it cannot provide any mobility. If the C&C server is not secure or mobile, it is easy to cleaned and ineffective. A single alarm or misuse report can provoke the C&C server to be quarantined or the botnet suspended.

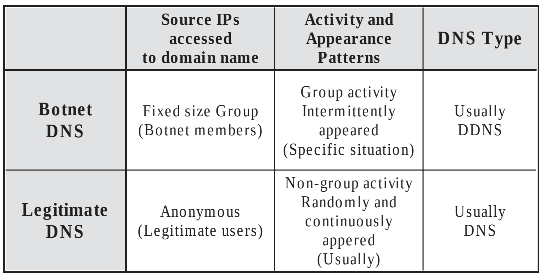
C. C&C Server Migration

If a botnet uses only a single C&C server, the botnet could easily be detected and disarmed. Thus, a botmaster wants to arrange several C&C servers which can be listed in the bot binary for the stability of the botnet and uses a dynamic DNS (DDNS) [15] which is a resolution service that automatically perceives the change of the IP address of a server and substitutes the DNS record by frequent updates and changes, for keeping the botnets portable. And even though the root C&C server cannot operate well or link failure occurred, candidate C&C servers can be a feasible substitution for the root C&C server.

It is observed that botnets were migrate their C&C server frequently [6], either by being instructed to move to a new IRC channel/server or to download a replacement software which pointed them to a different C&C server. There are some captured evidence of such migration occurrence which is simultaneously participating in two separate botnets. The botmaster move his botnet by changing the C&C server for evading to be captured. In the wild, there observed most of them (65%) are moved only up for 1 day [16]. Even though previous domain name of botnet C&C server is blocked, botmaster can just moves his botnet to another candidate C&C server.

D. Features of Botnet DNS

As mentioned above, infected hosts automatically access the C&C server with its domain name. Therefore, DNS RR (resource record) query is used and such a query also appears at other situations. Following 5 cases show the situations of the DNS query used in botnet. (1) At the rallying procedure: If the host infection success, the infected hosts should be gathered and as referred in previous section 3.B, DNS is used. (2) At the malicious behaviors of a botnet: Several types of malicious activities such as DDoS attack and spam mailing are accompanied with the DNS transmit. (3) At C&C server link failures: If the network or link of C&C server fails, bots cannot access to the C&C server, after a while (undergo failure of TCP 3-way handshaking), they begin to send the DNS query to DNS server. (4) At C&C server migration: As mentioned Section 3.C, the botnet migrate one to another candidate C&C server. In that moment, DNS query also used. (5) At C&C server IP address changes: If a C&C server uses dynamic allocated IP (DHCP), the corresponding IP address can be changed at any time and a botmaster also can change the IP address of the C&C server intentionally. If the IP address of the C&C server changed, the bots cannot connect the old IP address of the server, so they send the DNS query to access new C&C server.



	Source IPs accessed to domain name	Activity and Appearance Patterns	DNS Type
Botnet DNS	Fixed size Group (Botnet members)	Group activity Intermittently appeared (Specific situation)	Usually DDNS
Legitimate DNS	Anonymous (Legitimate users)	Non-group activity Randomly and continuously appered (Usually)	Usually DNS

Fig. 1. Differences between Botnet and Legitimate DNS

DNS queries of botnets can be distinguishable from legitimate DNS queries, by unique features of the botnet DNS queries. Figure 1 shows some differences between botnet DNS queries and legitimate DNS queries. First, only botnet members send queries to the domain name of C&C server(fixed size), legitimate user never queries to the C&C server domain name. Therefore, the number of different IP address which queried botnet domain is normally fixed. On the other hand, the legitimate cites are queried from anonymous users (random) at usually. Second, the fixed members of botnet act and migrate together at the same time. The group activity of botnet derived from this property. DNS queries from botnet occurr temporary and simultaneously. However, most of legitimate DNS queries occur continuously and do not occur simultaneously. The botnet queries appears at specified situations which mentioned above, so they appeared intermittently. Third, the botnet uses DDNS for C&C server usually, but legitimate cites do not commonly use DDNS.

IV. DNS-BASED BOTNET DETECTION MECHANISM

A. Botnet DNS Query Detection Algorithm

We developed a botnet DNS query detection algorithm by using the different features of botnet DNS and legitimate DNS which mentioned in Section 3.D. The algorithm separated 3 different parts which are (1) Insert-DNS-Query, (2) Delete-DNS-Query, (3) Detect-BotDNS-Query. Figure2 shows the Insert-DNS-Query stage of algorithm. There is a database for storing DNS query data which include source IP address of the query, domain name of the query and timestamp of the query received. We grouping the DNS query data by the domain name and timestamp. Fig 3, 4, 5 demonstrate the algorithm with pseudo code. First, there is an array A prepared for storing

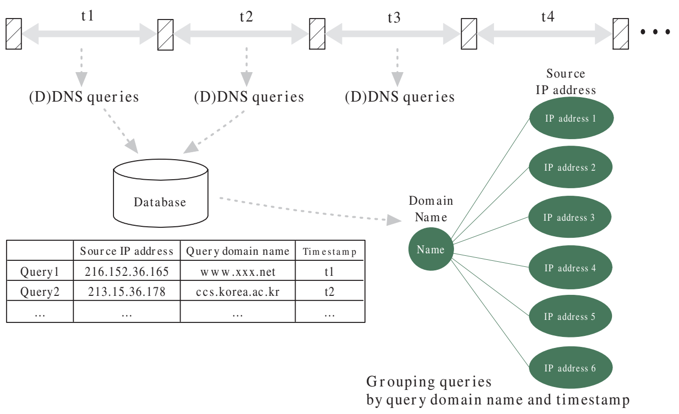
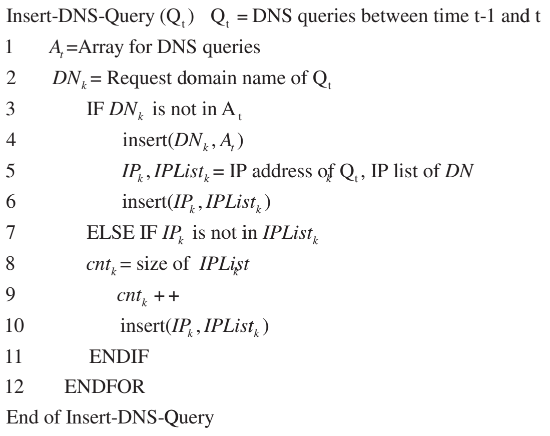
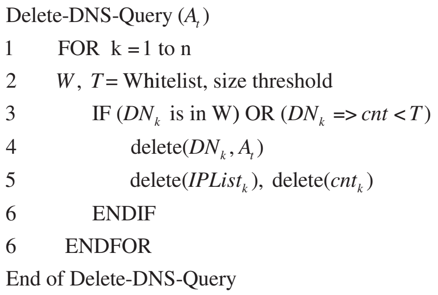


Fig. 2. Insert-DNS-Query



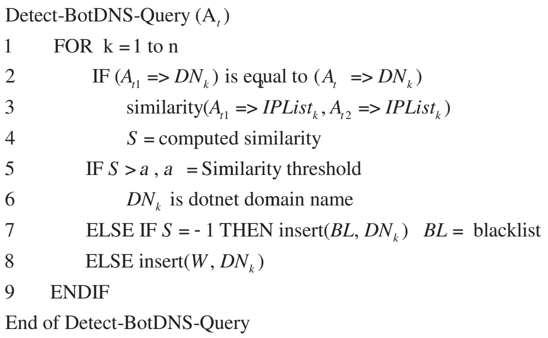
```
Insert-DNS-Query (Q_t)   Q_t = DNS queries between time t-1 and t
  $A_t$=Array for DNS queries
  $DN_k$ = Request domain name of $Q_t$
    IF $DN_k$ is not in $A_t$
        insert($DN_k$, $A_t$)
        $IP_k$, $IPList_k$ = IP address of $Q_t$, IP list of $DN_k$
        insert($IP_k$, $IPList_k$)
    ELSE IF $IP_k$ is not in $IPList_k$
    $cnt_k$ = size of $IPList_k$
        $cnt_k$ ++
        insert($IP_k$, $IPList_k$)
    ENDIF
  ENDFOR
End of Insert-DNS-Query
```

Fig. 3. Insert-DNS-Query



```
Delete-DNS-Query ($A_t$)
  FOR k = 1 to n
  $W$, $T$ = Whitelist, size threshold
    IF ($DN_k$ is in W) OR ($DN_k$ => $cnt < T$)
        delete($DN_k$, $A_t$)
        delete($IPList_k$), delete($cnt_k$)
    ENDIF
  ENDFOR
End of Delete-DNS-Query
```

Fig. 4. Delete-DNS-Query



```
Detect-BotDNS-Query ($A_t$)
  FOR k = 1 to n
    IF ($A_{t1}$ => $DN_k$) is equal to ($A_t$ => $DN_k$)
        similarity($A_{t1}$ => $IPList_k$, $A_{t2}$ => $IPList_k$)
        $S$ = computed similarity
    IF $S > a$ , $a$ = Similarity threshold
        $DN_k$ is dotnet domain name
    ELSE IF $S$ = - 1 THEN insert($BL$, $DN_k$)   $BL$ = blacklist
    ELSE insert($W$, $DN_k$)
  ENDIF
End of Detect-BotDNS-Query
```

Fig. 5. Detect-BotDNS-Query

the DNS queries. We inserted the domain name and source IP address of queries to A. If a new query comes in, checking it already existed in A. If it is a new domain name, insert data. Otherwise, check the IP address already exist in the IP list of the domain name and insert the IP address if it is not exist in the IP list. In this step, the data (domain name, source IP addresses and timestamps) of DNS queries are arranged by the requested domain name. Second, excute the Delete-DNS-Query step for removing redundant DNS query. If the size of IP list do not exceed the size threshold or the domain name is legitimate which already exist in a whitelist, the domain name of queries do not have to be processed. Therefore, it should be removed from array A for reducing the processing overhead and saving the memory. Finally, we find the botnet DNS queries in Detect-BotDNS-Query step. We define and compute numerical value of group activity of botnet DNS, called similarity. If there are two IP lists which are requested at time $t1$ and $t2$ and have a same domain name query, assume that each size of IP lists as A and B. And if there were same IP addresses between two IP lists, assume the size of duplicated IP addresses as C. We let S denote the similarity such that

$$S = \frac{1}{2} \cdot (\frac{C}{A} + \frac{C}{B})(A \neq 0, B \neq 0).$$

If $A = 0$ or $B = 0$ then we define the similarity as -1. If the similarity approximated 0, whitelisting the domain name and delete the IP list of the domain. Assume that there is domain name DN which requested by multiple source IP addresses in a certain time t, we measure how many source IP addresses of them request DN after t in each time slot. Due to the features of botnet DNS which mentioned in Section 3.4 the similarity of botnet DNS close to 1 different from legitimate DNS. And the suspicious domain name that occurred just one time and could be occurred later, which have the value of similarity -1, insert the domain name to blacklist to be monitored after that time.

B. Migrating Botnet Detection Algorithm

The algorithm of botnet DNS query detection enables us to distinguish the botnet. However, the algorithm cannot detect botnets migrating to another C&C server. Therefore, we developed the migrating botnet detection algorithm with modifying the botnet DNS query detection algorithm. The first and second stage(Insert-DNS-Query and Delete-DNS-Query) are same but third step of algorithm is different. During the migration of botnet, bots use two different domain name of C&C server, therefore we compare the IP lists of different domain name which have similar size of IP list. Here, similar size determined on basis of experiment. As we mentioned

Section 3.3, botmaster move their botnets frequently to change the C&C server and most of them (65%) are only up for 1 day in the wild. Therefore, the detection algorithm of migration activity is significant part of the botnet detection system.

C. Botnet Detection System

The botnet detection system that combines both of botnet query detection and migrating botnet detection, requires DNS traffic data. And it can be ideal that large scale of DNS traffic data from deployed sensors is provided for the input data because botnets usually dispersed at different networks. Therefore, if the detection system applied for small network, detection accuracy can be decreased. Moreover, the system is sensitive to the threshold values so, it must be carefully decided.

V. Evaluation

In order to evaluate the effectiveness of the proposed mechanism, we have measured the detection performance in our testbed network. The proposed mechanism is implemented as a botnet detection system and the system is executed on a campus network with botnets. We have created a Agobot code which is one of the most famous bot and secured the IRC C&C server and its channels. Over 50 machines are used in the botnet and the testbed network is linked with the campus network, therefore we carefully made our botnet invisible and secure to prevent botnet from being exposed. We made the scenario script for verifying the algorithms and the scenario includes botnet construction, rally to the C&C server and command and control for spam mailing, DDoS attack, C&C server migration, etc. The scenario contains the situation which mentioned in Section 3.4 for validating botnet DNS query detection algorithm. We also migrates our botnet from root IRC C&C server to candidate IRC server for verifying the migrating botnet detection algorithm. We use Pentium 4 processor PCs that operate on Windows XP. Default values of parameters are as follows. A time unit is 1 hour and a size threshold for the detection algorithm is 5(size of IP List) and similarity threshold is 0.8, because it is the adequate value which is between a similarity of botnet domain and a maximum similarity of legitimate domains. We tested our botnet for evaluation, and captured the traffic for 10 hours.

A. Botnet DNS Query Detection

The botnet in our testbed performs several kinds of activities which include spam mailing, DDoS, C&C server migration, etc. To be sure, some of them provoke DNS traffic and consequently, our algorithm can detect the botnet nicely. The size of IP address list are distributed as shown in Fig. 6. The size of IP list means the different number of source IP addresses which queried same domain name during 1 hour and the Fig. 6. shows that over 80% of the IP list size was 1. it means that most of the DNS queries are sent from only 1 host during 1 hour. The size threshold of IP list is settled with 5 and it results 92.5% of DNS queries eliminated which gives great efficiency of the botnet DNS query detection algorithm.

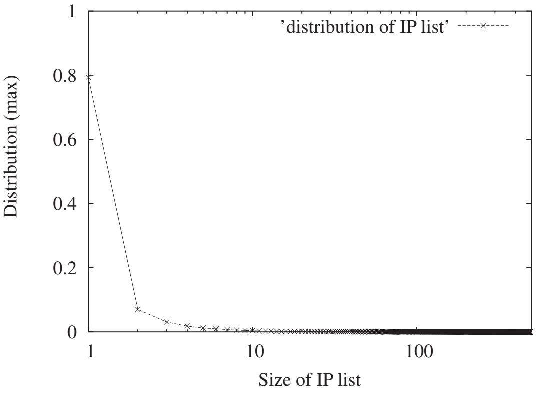


Fig. 6. Distribution of IP List Size

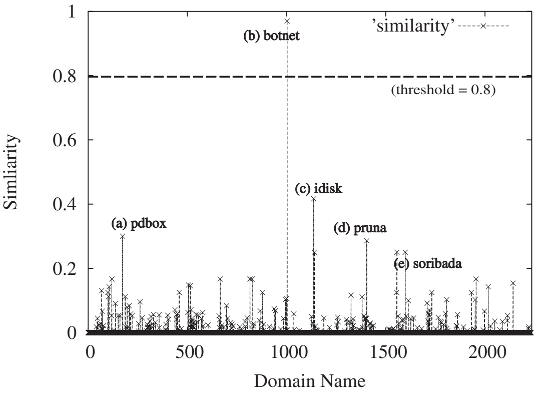


Fig. 7. Similarity of Each Domain Name

In conclusion, our algorithm can detect the botnet properly if over 5 members of botnet are existed in the C class size of network (the size of our experiment campus network). The algorithm check all domain names that was not eliminated from previous step. The similarities in a certain time t are shown in Fig. 7 and there are about 2300 different domain names which include botnet domain name (if the domain name could affiliated each other we plot the highest value of similarity). Most of similarities equal to 0 or -1 (90%). Suppose that domain name DN is source IP list A during time t and IP list B during $t+1$ queried DN. In that case, if a computed similarity of DN is equal to 0 and that means the IP List A are totally different from B. If the similarity of DN is -1, DN is just only requested just once (t or $t+1$) and they added in blacklist of the algorithm because they are suspicious to be the domain of botnet. Other domain names mostly ranged from 0 to 0.2 (7.4%). It implies that a certain host which queried a domain(ranged from 0 to 0.2) in timeslot t1, could send query to the same domain in t1+1 with the probability from 0% to 20%. Only the similarity of botnet domain exceeds threshold 0.8, so the botnet domain name could be detected. Some interesting domain names which have a similarity larger than 0.2 are shown in Fig. 7 ((a) (e)) and all of them were identified as P2P cites or a cite of enormous size of file transferring. (a) is the domain name of pdbox [17] and (c) is the domain name of idisk [18], both cites provide the service of uploading and downloading large size of personal files which are movie, game, mp3, etc. (d) is the domain name of pruna [19] and (e) is the domain name of soribada [20], both provide P2P service. We conjecture the reason that the users who have accessed P2P or file transferring cite tend to keep up the connection and more continuously access the same cite more than other cites. Therefore, the similarity of these domains have higher similarity than other domains.

B. Migrating Botnet Detection

We also run migrating bot detection algorithm with the scenario script. In the worst case, algorithm runs on $O(n^2)$. Nevertheless, our algorithm operates in a reasonable time (about 5 minute for 1 hour DNS trace) because the algorithm remove the set of IP lists which do not exceed the size threshold (92.5% of DNS queries removed). Here, the "similar size" are settled within 10% of the size of IP list. For example if the size of IP list is 100, then we compare the IP list with all of other IP list that has the size within 95 to 105. One of the results which include botnet migration is shown in Fig. 8 and the algorithm detect the migrating bot correctly. Most of IP list has the similarity that getting lower as the size of IP list increase, because if the size of IP list getting larger, a probability of which the source IP addresses between two similar size of IP list duplicates getting lower.

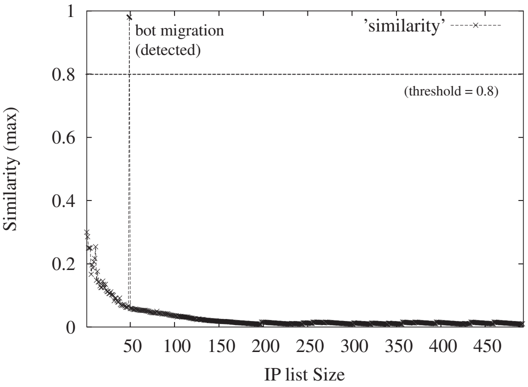


Fig. 8. Similarity of Each Size of IP List

VI. Discussion

Our algorithm worked properly in reasonable processing time, but if we assume the situation that our system monitor huge scale of network then the processing time can be a big problem. Hash tables are a great solution for dealing the IP address lookup and we consider it for our future work.

The botnet can evade our algorithms when the botnet uses DNS only at initializing and never use it again (moreover, do not migrate the botnet). If we could find IP group list of IRC traffic in C&C process or attack traffic such as spam mailing or DDoS attack, we can compare each IP lists of them. Here, the IP lists provider can be the IDS, IPS or other attack detection systems.

It is possible to paralyze our algorithm with intentionally generated DNS queries that spoof their sources. The fabricated packets, our algorithm could be poisoned. In this research, we do not care about the situation of poisoning, but a simple preprocessing can be a solution. If we check the 3-way handshaking of TCP traffic and record the IP addresses to the list which endures handshaking. Then we could eliminates the faked IP addresses of the DNS traffic that do not endure the handshaking.

VII. Conclusion

It is necessary to provide appropriate countermeasure for botnet which become a one of the biggest threat of network security and major contributor to unwanted network traffic. Therefore we researched a simple mechanism to detect a botnet by using a DNS queries which used by botnet. We found significant features of botnet DNS queries which discriminate from legitimate DNS queries. The two different algorithm for botnet detection are proposed and both can detect the specific activity of botnet nicely. With our suggested system network administrator enable to detect bot agents and dispose them.

References

[1] J. Oikarinen and D. Reed, "Internet relay chat protocol." RFC 1459, 1993.

[2] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Internet Measurements Conference (IMC '06)*, Oct 2006.

[3] J. Jones, "Botnets: Detection and mitigation," Feb 2003. FEDCIRC.

[4] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understanding, detecting, and disturbing botnets," in *The 1st Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI '05)*, July 2005.

[5] P. Barford and V. Yegneswaran, "An inside look at botnets," 2006. Special Workshop on Malware Detection, Advances in Information Security, Springer Verlag.

[6] D. Dagon, C. Zou, and W. Lee, "Modeling botnet propagation using time zones," in *NDSS 2006*, Feb 2006.

[7] D. Dagon, "Botnet detection and response," in *OARC Workshop, 2005*, 2005.

[8] J. Kristoff, "Botnets," Oct 2004. 32nd Meeting of the North American Network Operators Group.

[9] J. R. Binkley and S. Singh, "An algorithm for anomaly-based botnet detection," in *The 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI '06)*, 2006.

[10] A. Ramachandran, N. Feamster, and D. Dagon, "Revealing botnet membership using dnsbl counter-intelligence," in *The 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI '06)*, 2006.

[11] M. Overton, "Bots and botnets," in *Virus Bulletin 2005*, Oct 2005.

[12] "Ciphertrust, secure computing." http://www.ciphertrust.com/.

[13] Symantec Co., http://www.symantec.com/.

[14] D. McPherson, "Fingerprint sharing: The need for automation of inter-domain information sharing," May 2005. http://www.arbornetworks.com/.

[15] P. Vixie, S. Thomson, Y. Rekhter, , and J. Bound, "Dynamic updates in the domain name system (dns update)," 1997. http://www.faqs.org/rfcs/rfc2136.html/.

[16] D. Song, "personal communication," Oct 2006. Korea University Security Seminar.

[17] NOWCOM Co., pdbox, http://www.pdbox.co.kr/.

[18] KTH Co., idisk, http://idisk.paran.com/.

[19] MEDIAPORT Co., pruna P2P, http://www.pruna.com/.

[20] SORIBADA Inc., soribada P2P, http://www.soribada.com/.